

Relational Efficiencies: Part I

Renee Teatro
Information Builders
F.I.R.E. Conference
October 2008

Relational Efficiencies

Agenda

- Optimization Overview
- JOIN Scenarios
- Sort Scenarios

Relational Efficiencies

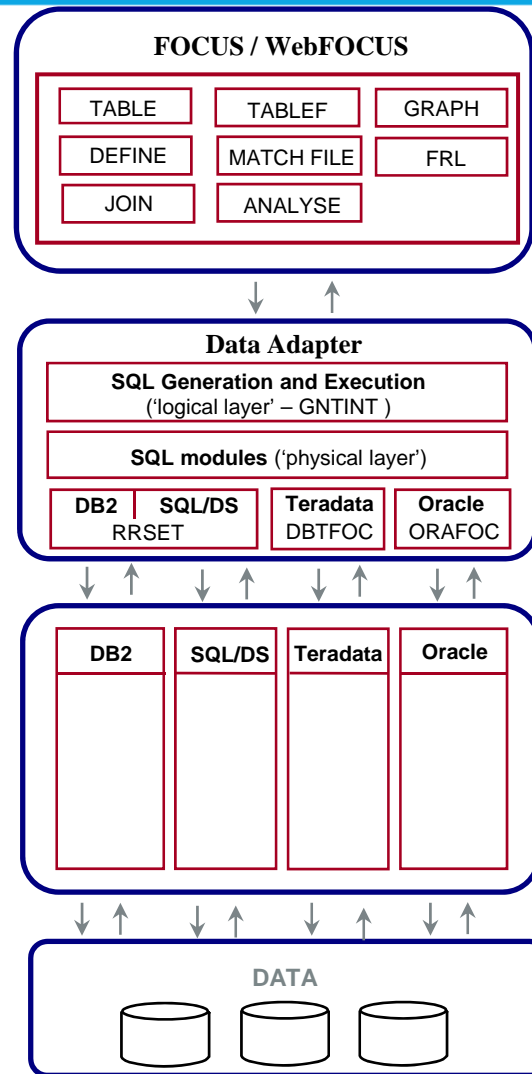
Layers of Processing

User executes a FOCUS request...

- FOCUS calls the appropriate module (TABLE, GRAPH)
- Reads and parses the MFD
- Parses the request
- Calls the Interface with information on the MFD and the request available in work areas

- Reads and parses the AFD
- Checks the module specifics for optimization
- Analyzes and optimizes the request
- Generates SELECT statement(s)
- Passes SELECT statements to the appropriate 'physical' module
- Prepares, allocates, and opens cursor(s)
- Calls the RDBMS to FETCH rows of data

- Analyzes and optimizes the SQL translation of the FOCUS application request
- Chooses the appropriate access path and retrieval method
- Retrieves data
- Creates the answer set



Report is displayed

- FOCUS reads a row from the answer set and processes the remaining actions on that row (IF/WHERE, DEFINES...)
- Puts the valid row into the Internal Matrix
- Reads the next row and repeats the process until the end of the answer set
- Processes the Internal Matrix and displays the report

- The Interface receives a row and/or an SQL status code
- Converts non-standard data into FOCUS format, making it available to FOCUS
- Asks RDBMS for the next row (FETCH), until the end of the answer set is reached (SQL status code +100)

- Sends back a row and/or an SQL status code to the Interface

Relational Efficiencies

RDBMS Optimization

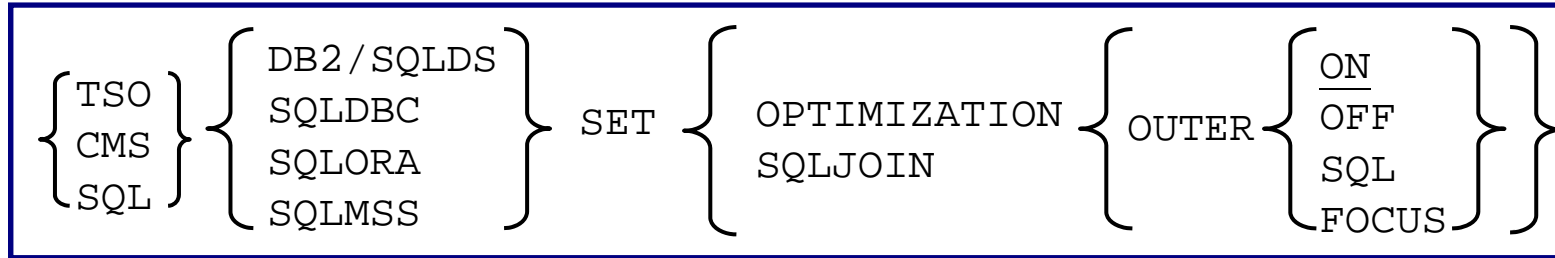
Interface optimization is the degree to which a TABLE request is translated to SQL

In other words, the process in which the interface translates projection, selection, JOIN, sort, and aggregation operations of a report request into its SQL equivalent and passes it to the RDBMS for processing

- TABLE, MODIFY, MAINTAIN → SQL
- Direct SQL Passthru
- Passing JOINS

Relational Efficiencies

The Optimization Command



Components	Description	
Target Database Engine	DB2 / SQLDS SQLDBC SQLORA SQLMSS	for DB2 for Teradata for Oracle for SQL Server

Relational Efficiencies

Optimization Settings

In the SQL *engine* SET OPTIMIZATION ON is the default, so the interface attempts to fully optimize all requests

- RDBMS is preferred engine for processing
- More RDBMS processing is usually beneficial

Types of optimization

- ON – default
- OFF – FOCUS / WebFOCUS will handle all JOINS, sorting and aggregation.
- SQL – pass SQL even if multiplicative effect found
- FOCUS – pass SQL only if results are identical to FOCUS / WebFOCUS processing

Relational Efficiencies

Data Adapter TRACE Facility

- SET TRACEON=component//destination
- Component
 - SQLDI - FSTRACE - All Adapter-RDBMS activity
 - SQLAGGR - FSTRACE3 - Optimization messages
 - STMTRACE - FSTRACE4 - SQL only
 - SQLCALL - commands and data exchange between the physical and the logical layers of the data adapter
- Destination
 - FSTRACE - allocation for the ddname of FSTRACE
 - CLIENT - displays client session to the screen

NOTE: Make sure to SET TRACEUSER=OFF afterwards

Relational Efficiencies

Data Adapter TRACE Facility

```
SET TRACEOFF=ALL
SET TRACEUSER=CLIENT
SET TRACEON=SQLAGGR//CLIENT
SET TRACEON=STMTRACE//CLIENT
SET TRACESTAMP=OFF
SET XRETRIEVAL=OFF
TABLE FILE ...
```

SET XRETRIEVAL=[ON | OFF]

- **ON** – the data adapter sends the request to the RDBMS and it processes the request
- **OFF** – the data adapter attempts to optimize the request and trace output is generated but no RDBMS processing is done

Relational Efficiencies

Fully Optimized Query

```
SET TRACEOFF=ALL
SET TRACEUSER=CLIENT
SET TRACEON=STMTRACE//FSTRACE
TABLE FILE EMPLOYEE
COUNT EMP_ID BY DEPARTMENT
WHERE CURR_SAL GT 10000
END
```



STMTRACE:

```
SELECT DEPARTMENT, COUNT(*) FROM EMPLOYEE
WHERE (CURR_SAL > 10000)
GROUP BY DEPARTMENT ORDER BY DEPARTMENT;
```

Relational Efficiencies: JOIN Scenarios

JOIN Optimization

- Interface attempts to generate ONE SELECT statement to JOIN all tables
- Applies to dynamic or embedded JOINS
 - One OPEN cursor operation
 - JOIN optimized more readily by RDBMS
 - An optimized JOIN enables sorts and aggregations to be passed
 - Limits interface \Leftrightarrow RDBMS communications
- If JOIN is not passed to RDBMS
 - Termed: FOCUS-managed JOIN
 - One SELECT statement for EACH table

Relational Efficiencies: JOIN Scenarios

Optimized JOIN vs. Non-Optimized JOIN

```
JOIN F1SSN IN TABLE1 TO F2SSN IN TABLE2  
TABLE FILE TABLE1  
PRINT F2SSN  
END
```

Optimized JOIN:

```
SELECT T1.F1SSN,T2.F2SSN FROM PROD.TABLE1 T1,  
      PROD.TABLE2 T2 WHERE (T2.F2SSN = T1.F1SSN)  
      FOR FETCH ONLY;
```

Non-Optimized JOIN:

(FOC2510) FOCUS-MANAGED JOIN SELECTED FOR FOLLOWING REASON(S):

```
SELECT T1.F1SSN FROM PROD.TABLE1 T1 FOR FETCH ONLY;  
SELECT T2.F2SSN FROM PROD.TABLE2 T2 WHERE (T2.F2SSN =  
?)    FOR FETCH ONLY;
```

Relational Efficiencies: JOIN Scenarios

Performance related to KEYS attribute

```
JOIN CLEAR *
JOIN AU_ID IN AUTHORS TO ALL AU_ID IN TITLEAUTHOR AS J1
TABLE FILE AUTHORS
SUM TITLE_ID AU_ORD ROYALTYPER
BY AU_LNAME
BY AU_FNAME
BY AU_ID
END
```

```
FOC2510 - FOCUS-MANAGED JOIN SELECTED FOR FOLLOWING REASON(S):
FOC2505 - A SEGMENT IN THE STRUCTURE IS NON-KEYED: : AUTHORS
FOC2517 - FST. OR LST. WHERE SORT FIELDS DO NOT COVER KEY
FOC2590 - AGGREGATION NOT DONE FOR THE FOLLOWING REASON:
FOC2592 - RDBMS-MANAGED JOIN HAS BEEN DISABLED
SELECT T1."au_id",T1."au_lname",T1."au_fname" FROM
pubs.dbo.authors T1;
SELECT T2."au_id",T2."title_id",T2."au_ord",T2."royaltyper"
FROM pubs.dbo.titleauthor T2 WHERE (T2."au_id" = ?) ORDER BY
T2."au_id",T2."title_id";
```

Relational Efficiencies: JOIN Scenarios

Performance related to KEYS attribute

The `.acx` (ACCESS FILE DESCRIPTION) should have `KEYS=` with a `NON-ZERO` value to indicate the PRIMARY KEY for the table in question. The PRIMARY KEY is made up of the FIRST `n` fields in the `.mas` (MASTER FILE DESCRIPTION).

```
SEGNAME=AUTHORS, TABLENAME=pubs.dbo.authors,  
  CONNECTION=SQLSRV, KEYS=0,$ *** Change to KEYS=1  
  
SEGNAME=TITLEAUTHOR, TABLENAME=pubs.dbo.titleauthor,  
  CONNECTION=SQLSRV, KEYS=2,$
```

AGGREGATION DONE ...

```
SELECT T1."au_lname",T1."au_fname",T1."au_id",  
MAX(T2."title_id"), SUM(T2."au_ord"), SUM(T2."royaltyper") FROM  
pubs.dbo.authors T1,pubs.dbo.titleauthor T2 WHERE (T2."au_id" =  
T1."au_id") GROUP BY T1."au_lname",T1."au_fname",T1."au_id"  
ORDER BY T1."au_lname",T1."au_fname",T1."au_id";
```

Relational Efficiencies: JOIN Scenarios

Performance related to KEYS attribute

Another MFD/AFD by the same name higher in the application path list with the SAME name as one of the JOINed tables.

```
WHENCE filename MASTER
```

```
WHENCE AUTHORS MASTER
```

```
WHENCE TITLEAUTHOR MASTER
```

```
C:\ibi\apps\sqlsrv\authors.mas
```

```
C:\ibi\apps\summit08\titleauthor.mas
```

```
APP PREPENDPATH summit08
```

```
WHENCE AUTHORS MASTER
```

```
WHENCE TITLEAUTHOR MASTER
```

```
C:\ibi\apps\summit08\authors.mas
```

```
C:\ibi\apps\summit08\titleauthor.mas
```

Relational Efficiencies: JOIN Scenarios

Performance related to MULTI-PATH

```
SET TRACEOFF=ALL
SET TRACEUSER=CLIENT
SET TRACEON=SQLAGGR//CLIENT
SET TRACEON=STMTRACE//CLIENT
SET TRACESTAMP=OFF
SET XRETRIEVAL=OFF
SET MULTIPATH=COMPOUND
JOIN CLEAR *
JOIN TITLE_ID IN TITLEAUTHOR TO ALL TITLE_ID IN TITLES AS J1
JOIN TITLE_ID IN TITLEAUTHOR TO ALL TITLE_ID IN ROYSCHED AS J2
CHECK FILE TITLEAUTHOR PICT
TABLE FILE TITLEAUTHOR
PRINT TITLE AU_ORD ROYALTYPER PRICE LORANGE HIRANGE
BY TITLE_ID
END
```

Relational Efficiencies: JOIN Scenarios

Performance related to MULTI-PATH - Optimized

```
FOC2590 - AGGREGATION NOT DONE FOR THE FOLLOWING REASON:  
FOC2594 - AGGREGATION IS NOT APPLICABLE TO THE VERB USED  
SELECT T1."title_id",T1."au_ord",T1."royaltyper",T2."lorange",  
T2."hirange",T3."title",T3."price" FROM pubs.dbo.titleauthor T1,  
pubs.dbo.roysched T2,pubs.dbo.titles T3 WHERE (T2."title_id" =  
T1."title_id") AND (T3."title_id" = T1."title_id") ORDER BY  
T1."title_id";
```

Prior Behavior

```
(FOC2510) FOCUS-MANAGED JOIN SELECTED FOR FOLLOWING REASON(S):  
(FOC2512) MULTI-PATH REQUEST  
(FOC2590) AGGREGATION NOT DONE FOR THE FOLLOWING REASON:  
(FOC2592) RDBMS-MANAGED JOIN HAS BEEN DISABLED
```

- Adjust JOIN field(s) to create a SINGLE PATH structure
- SQL Passthru
- Create relational VIEW on RDBMS - single segment MFD
- Create HOLD files & JOIN HOLD files vs. FOCUS-managed JOIN

Relational Efficiencies: JOIN Scenarios

Performance related to MULTI-PATH

CHECK FILE TITLEAUTHOR PICTURE

```
0 NUMBER OF ERRORS=      0
NUMBER OF SEGMENTS=    3 ( REAL=    1 VIRTUAL=    2 )
NUMBER OF FIELDS=    18 INDEXES=    0 FILES=    3
TOTAL LENGTH OF ALL FIELDS= 381
1SECTION 01
STRUCTURE OF GNTINT   FILE TITLEAUT ON 05/15/08 AT 19.06.20
TITLEAU>
01      S0
*****
*AU_ID      **
*TITLE_ID   **
*AU_ORD     **
*ROYALTYPER **
*          **
*****
*****
I
+-----+
I          I
I ROYSCHED I TITLES
02  I KM   03  I KM
.....
:TITLE_ID  ::K :TITLE_ID  ::K
:LORANGE   ::  :TITLE     ::
:HIRANGE   ::  :TYPE      ::
:ROYALTY   ::  :PUB_ID    ::
:          ::  :          ::
:          ::  :          ::
:          ::  :          ::
.....
JOINED ROYSCHED JOINED TITLES
```

Relational Efficiencies: JOIN Scenarios

Performance related to MULTI-PATH

```
JOIN CLEAR *  
JOIN TITLE_ID IN TITLEAUTHOR TO ALL TITLE_ID IN TITLES AS J1  
JOIN TITLES.TITLE_ID IN TITLEAUTHOR TO ALL TITLE_ID IN ROYSCHED AS J2
```

```
STRUCTURE OF GNTINT      FILE TITLEAUT ON 05/16/08 AT 14.07.27  
TITLEAU>  
01      S0  
*****  
*AU_ID      **  
*TITLE_ID   **  
*AU_ORD     **  
*ROYALTYPER **  
*           **  
*****  
I  
I  
I  
I TITLES  
02      I KM  
-----  
:TITLE_ID   : :K  
:TITLE      : :  
:TYPE       : :  
:PUB_ID     : :  
:           : :  
:           : :  
-----  
JOINED I TITLES  
I  
I  
I ROYSCHED  
03      I KM  
-----  
:TITLE_ID   : :K  
:LORANGE    : :  
:HIRANGE    : :  
:ROYALTY    : :  
:           : :  
:           : :  
-----  
JOINED ROYSCHED
```

Benefits of Single Path:

Avoid encounter with FOC030

(FOC030) ALL VERB OBJECTS MUST
BE IN THE SAME PATH AS THEIR
SORT FIELDS

Relational Efficiencies: JOIN Scenarios

Performance related to MULTI-PATH

```
JOIN CLEAR *
```

```
JOIN TITLE_ID IN TITLEAUTHOR TO TITLE_ID IN TITLES AS J1
```

```
JOIN TITLE_ID IN TITLEAUTHOR TO ALL TITLE_ID IN ROYSCHED AS J2
```

```
0 NUMBER OF ERRORS=      0
NUMBER OF SEGMENTS=    3 ( REAL=    1 VIRTUAL=    2 )
NUMBER OF FIELDS=     18 INDEXES=    0 FILES=    3
TOTAL LENGTH OF ALL FIELDS= 381
1SECTION 01
STRUCTURE OF GNTINT      FILE TITLEAUT ON 05/16/08 AT 14.21.15
TITLEAU>
01      S0
*****
*AU_ID      **
*TITLE_ID   **
*AU_ORD     **
*ROYALTYPER **
*          **
*****
*****
I
+-----+
I          I
I TITLES   I ROYSCHED
02      I KU      03      I KM
:
: TITLE_ID :K      : TITLE_ID  ::K
: TITLE    :      : LORANGE   ::
: TYPE     :      : HIRANGE   ::
: PUB_ID   :      : ROYALTY   ::
:          :      :          ::
:          :      :          ::
:          :      :          ::
JOINED TITLES .....
JOINED ROYSCHED
```

Relational Efficiencies: JOIN Scenarios

Multiplicative Effect

```
JOIN CLEAR *
JOIN TITLE_ID IN TITLES TO ALL TITLE_ID IN TITLEAUTHOR AS J1
TABLE FILE TITLES
SUM ROYALTY AU_ORD ROYALTYPER
BY TITLE
BY TITLE_ID
END
```

```
FOC2506 - INTERFACE-MANAGED NATIVE JOIN SELECTED FOR THE FOLLOWING REASON:
FOC2516 - MULTIPLICATIVE EFFECT WOULD ENSUE
FOC2590 - AGGREGATION NOT DONE FOR THE FOLLOWING REASON:
FOC2592 - RDBMS-MANAGED JOIN HAS BEEN DISABLED
SELECT T1."title_id",T1."title",T1."royalty",T2."au_id",
T2."title_id",T2."au_ord",T2."royaltyper" FROM pubs.dbo.titles
T1,pubs.dbo.titleauthor T2 WHERE (T2."title_id" =
T1."title_id") ORDER BY T1."title_id",T2."au_id",T2."title_id";
```

Relational Efficiencies: JOIN Scenarios

Multiplicative Effect

```
JOIN CLEAR *
JOIN TITLE_ID IN TITLES TO ALL TITLE_ID IN TITLEAUTHOR AS J1
TABLE FILE TITLES
SUM MAX.ROYALTY AU_ORD ROYALTYPER
BY TITLE
BY TITLE_ID
END
```

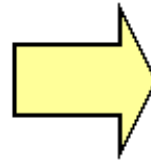
AGGREGATION DONE ...

```
SELECT T1."title",T1."title_id", MAX(T1."royalty"),
SUM(T2."au_ord"), SUM(T2."royaltyper") FROM pubs.dbo.titles T1,
pubs.dbo.titleauthor T2 WHERE (T2."title_id" = T1."title_id")
GROUP BY T1."title",T1."title_id" ORDER BY T1."title",
T1."title_id";
```

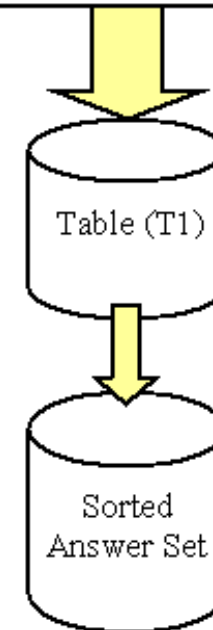
Relational Efficiencies: Sort Scenarios

Sort Management by the RDBMS

```
TABLE FILE EMPINFO  
PRINT FIRST_NAME DEPARTMENT  
      ED_HRS  
BY HIRE_DATE  
BY LAST_NAME  
WHERE ED_HRS GT 10  
END
```



```
SELECT HIRE_DATE, LAST_NAME,  
FIRST_NAME, DEPARTMENT,  
ED_HRS FROM EMPINFO T1  
ORDER BY HIRE_DATE,  
LAST_NAME, WHERE ED_HRS >  
10;
```



Relational Efficiencies: Sort Scenarios

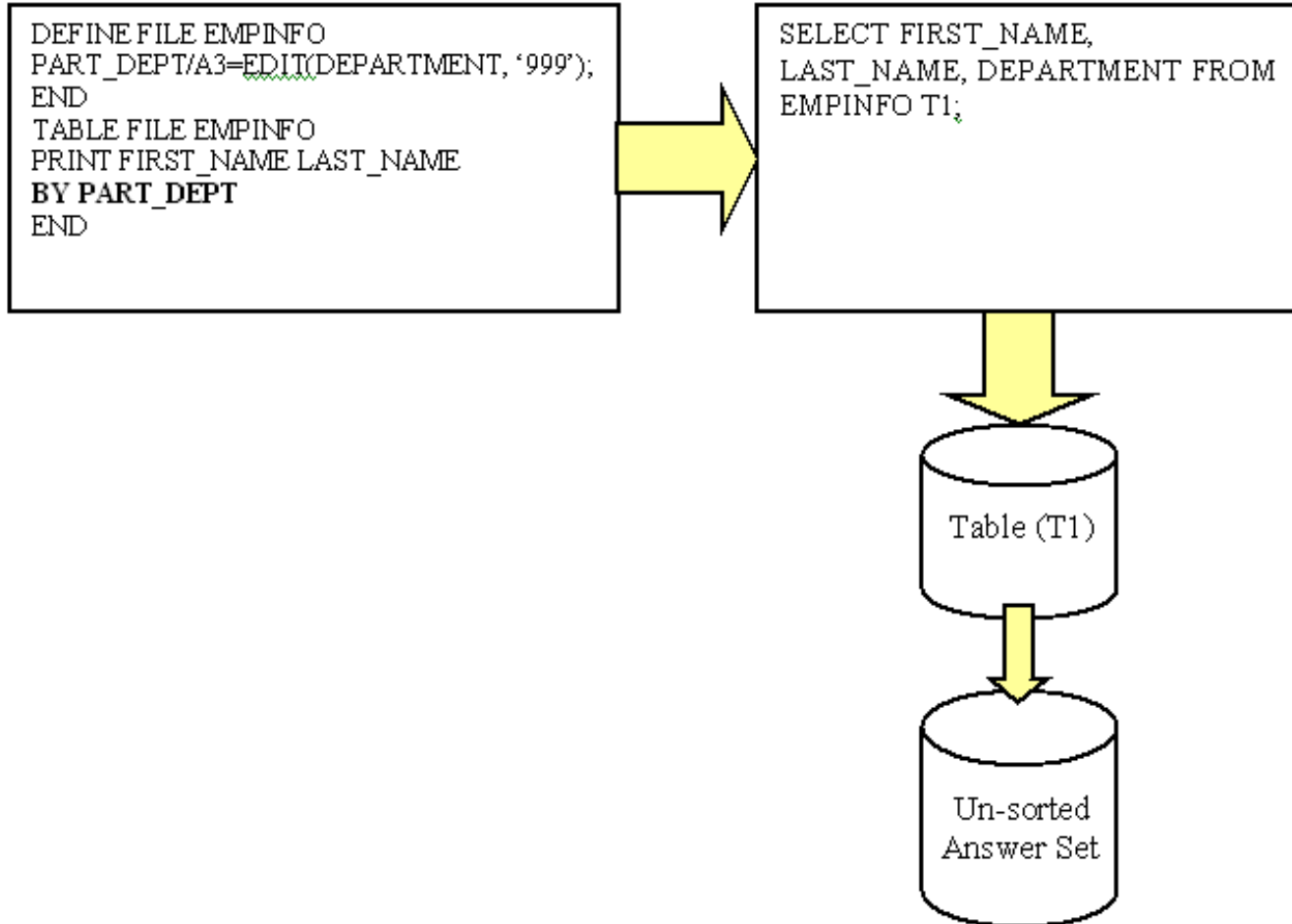
Sort Management by the RDBMS

- BY and ACROSS equate to SQL ORDER BY clause
- BY / FOR ROWS OVER syntax will translate

```
JOIN TITLE_ID IN TITLES TO ALL TITLE_ID IN ROYSCHED AS J1
TABLE FILE TITLES
SUM ADVANCE LORANGE HIRANGE BY PUB_ID
FOR TITLE_ID
'PS2091' OVER
'PS2106' OVER
'MC3021' OVER
'BU1111' OVER
'PC8888'
END
AGGREGATION DONE ...
SELECT T1."pub_id",T1."title_id", SUM(T1."advance"),
SUM(T2."lorange"), SUM(T2."hirange") FROM pubs.dbo.titles T1,
pubs.dbo.roysched T2 WHERE (T2."title_id" = T1."title_id") AND
(T1."title_id" IN('BU1111', 'MC3021', 'PC8888', 'PS2091',
'PS2106')) GROUP BY T1."pub_id",T1."title_id" ORDER BY
T1."pub_id",T1."title_id";
```

Relational Efficiencies: Sort Scenarios

Sort Management by FOCUS / WebFOCUS



Relational Efficiencies: Sort Scenarios

Sort Management by FOCUS / WebFOCUS

```
DEFINE FILE TITLES
```

```
SHORT_TITLE/A10=EDIT(TITLE, '9999999999');
```

```
PRICE_INC/P21.2=PRICE * .10;
```

```
END
```

```
TABLE FILE TITLES
```

```
PRINT PRICE
```

```
BY TITLE_ID
```

```
BY PRICE_INC
```

```
-*BY SHORT_TITLE
```

```
END
```

```
FOC2590 - AGGREGATION NOT DONE FOR THE FOLLOWING REASON:
```

```
FOC2594 - AGGREGATION IS NOT APPLICABLE TO THE VERB USED
```

```
SELECT T1."title_id",T1."price" FROM pubs.dbo.titles T1;
```

```
FOC2590 - AGGREGATION NOT DONE FOR THE FOLLOWING REASON:
```

```
FOC2594 - AGGREGATION IS NOT APPLICABLE TO THE VERB USED
```

```
SELECT T1."title_id",T1."title",T1."price" FROM pubs.dbo.titles T1;
```

Relational Efficiencies: Sort Scenarios

Sort Management by the RDBMS

```
DEFINE FILE TITLES
```

```
SHORT_TITLE/A10=EDIT(TITLE, '9999999999');
```

```
PRICE_INC/P21.2=PRICE * .10;
```

```
END
```

```
TABLE FILE TITLES
```

```
SUM PRICE BY TITLE_ID
```

```
BY PRICE_INC
```

```
-*BY SHORT_TITLE
```

```
END
```

```
AGGREGATION DONE ...
```

```
SELECT T1."title_id", (T1."price" * .1), SUM(T1."price") FROM
```

```
pubs.dbo.titles T1 GROUP BY T1."title_id", (T1."price" * .1)
```

```
ORDER BY T1."title_id", (T1."price" * .1);
```

```
AGGREGATION DONE ...
```

```
SELECT T1."title_id", (SUBSTRING(T1."title", 1, 10)),
```

```
SUM(T1."price") FROM pubs.dbo.titles T1 GROUP BY T1."title_id",
```

```
(SUBSTRING(T1."title", 1, 10)) ORDER BY T1."title_id",
```

```
(SUBSTRING(T1."title", 1, 10));
```

Relational Efficiencies: Sort Scenarios

Sort Management by the RDBMS

```
DEFINE FILE TITLES
```

```
MONTH_COMPONENTA/A3=HNAME(PUBDATE, 'MONTH', 'A3');
```

```
MONTH_COMPONENTI/I2=HPART(PUBDATE, 'MONTH', 'I2');
```

```
END
```

```
TABLE FILE TITLES
```

```
SUM PRICE
```

```
-*BY MONTH_COMPONENTA
```

```
BY MONTH_COMPONENTI
```

```
-*BY PUBDATE
```

```
END
```

Tracing via DEFINE field MONTH_COMPONENTI or MONTH_COMPONENTA sort:

```
FOC2590 - AGGREGATION NOT DONE FOR THE FOLLOWING REASON:
```

```
FOC2609 - CANNOT AGGREGATE BY NON-SQLIZABLE SORT KEY: MONTH_COMPONENTI
```

```
SELECT T1."price",T1."pubdate" FROM pubs.dbo.titles T1;
```

Tracing via REAL field ORD_DATE sort:

```
AGGREGATION DONE ...
```

```
SELECT T1."pubdate", SUM(T1."price") FROM pubs.dbo.titles T1
```

```
GROUP BY T1."pubdate" ORDER BY T1."pubdate";
```

Relational Efficiencies: Sort Scenarios

Sort Management by the RDBMS

TABLE FILE TITLES

SUM PRICE

-*BY MONTH_COMPONENTA

BY MONTH_COMPONENTI

-*BY PUBDATE

END

AGGREGATION DONE ...

```
SELECT T1."pubdate", SUM(T1."price") FROM pubs.dbo.titles T1
GROUP BY T1."pubdate" ORDER BY T1."pubdate";
```

FILENAME=TITLES, SUFFIX=SQLMSS , \$

SEGMENT=TITLES, SEGTYPE=S0, \$

FIELDNAME=TITLE_ID, ALIAS=title_id, USAGE=A6, ACTUAL=A6, \$

FIELDNAME=TITLE, ALIAS=title, USAGE=A80, ACTUAL=A80, \$

...

FIELDNAME=PUBDATE, ALIAS=pubdate, USAGE=HYYMDs, ACTUAL=HYYMDs, \$

FIELDNAME=MONTH_COMPONENTI, ALIAS=pubdate, USAGE=M, ACTUAL=DATE, \$

FIELDNAME=MONTH_COMPONENTA, ALIAS=pubdate, USAGE=MT, ACTUAL=DATE, \$

Relational Efficiencies: Sort Scenarios

Using TABLEF

- From a performance standpoint, consider using TABLEF instead of TABLE for requests that you know are being returned in the requested sort order. This will free FOCUS / WebFOCUS from having to verify the sort order by loading its internal matrix. FOCUS / WebFOCUS will act directly on the data returned by the RDBMS.
- ACROSS phrases are not supported.
- It must be verified that an ORDER BY statement has been generated in the SQL for the correct columns prior to using TABLEF.
- Multi-verb requests are not supported.
- RETYPE is not available.
- TABLEF can hold locks on data pages until the data adapter issues a COMMIT (usually when the report displays). Locks may prevent access to the data by other applications. (online core FOCUS)
- Certain prefix operators are not available (TOT., PCT.)
- Computes that use prefix operators are not supported.

Relational Efficiencies

Thank You !

Relational Efficiencies

Appendix

```
FILENAME=EMPLOYEE, SUFFIX=SQLMSS , $
SEGMENT=EMPLOYEE, SEGTYPE=S0, $
  FIELDNAME=EMP_ID, ALIAS=emp_id, USAGE=A9, ACTUAL=A9, $
  FIELDNAME=FNAME, ALIAS=fname, USAGE=A20, ACTUAL=A20, $
  FIELDNAME=MINIT, ALIAS=minit, USAGE=A1, ACTUAL=A1,
  MISSING=ON, $
  FIELDNAME=LNAME, ALIAS=lname, USAGE=A30, ACTUAL=A30, $
  FIELDNAME=JOB_ID, ALIAS=job_id, USAGE=I6, ACTUAL=I2, $
  FIELDNAME=JOB_LVL, ALIAS=job_lvl, USAGE=I6, ACTUAL=I4,
  MISSING=ON, $
  FIELDNAME=PUB_ID, ALIAS=pub_id, USAGE=A4, ACTUAL=A4, $
  FIELDNAME=HIRE_DATE, ALIAS=hire_date, USAGE=HYYMDs, ACTUAL=HYYMDs, $

SEGNAME=EMPLOYEE, TABLENAME=pubs.dbo.employee, CONNECTION=SQLSRV, KEYS=1,$
```

Relational Efficiencies

Appendix

```
FILENAME=AUTHORS, SUFFIX=SQLMSS, $
SEGMENT=AUTHORS, SEGTYPE=S0, $
  FIELDNAME=AU_ID, ALIAS=au_id, USAGE=A11, ACTUAL=A11, $
  FIELDNAME=AU_LNAME, ALIAS=au_lname, USAGE=A40, ACTUAL=A40, $
  FIELDNAME=AU_FNAME, ALIAS=au_fname, USAGE=A20, ACTUAL=A20, $
  FIELDNAME=PHONE, ALIAS=phone, USAGE=A12, ACTUAL=A12, $
  FIELDNAME=ADDRESS, ALIAS=address, USAGE=A40, ACTUAL=A40,
  MISSING=ON, $
  FIELDNAME=CITY, ALIAS=city, USAGE=A20, ACTUAL=A20,
  MISSING=ON, $
  FIELDNAME=STATE, ALIAS=state, USAGE=A2, ACTUAL=A2,
  MISSING=ON, $
  FIELDNAME=ZIP, ALIAS=zip, USAGE=A5, ACTUAL=A5,
  MISSING=ON, $
  FIELDNAME=CONTRACT, ALIAS=contract, USAGE=I11, ACTUAL=I4, $

SEGNAME=AUTHORS, TABLENAME=pubs.dbo.authors, CONNECTION=SQLSRV, KEYS=1,$
```

Relational Efficiencies

Appendix

```
FILENAME=TITLES, SUFFIX=SQLMSS , $
SEGMENT=TITLES, SEGTYPE=S0, $
  FIELDNAME=TITLE_ID, ALIAS=title_id, USAGE=A6, ACTUAL=A6, $
  FIELDNAME=TITLE, ALIAS=title, USAGE=A80, ACTUAL=A80, $
  FIELDNAME=TYPE, ALIAS=type, USAGE=A12, ACTUAL=A12, $
  FIELDNAME=PUB_ID, ALIAS=pub_id, USAGE=A4, ACTUAL=A4,
  MISSING=ON, $
  FIELDNAME=PRICE, ALIAS=price, USAGE=P21.4, ACTUAL=P10,
  MISSING=ON, $
  FIELDNAME=ADVANCE, ALIAS=advance, USAGE=P21.4, ACTUAL=P10,
  MISSING=ON, $
  FIELDNAME=ROYALTY, ALIAS=royalty, USAGE=I11, ACTUAL=I4,
  MISSING=ON, $
  FIELDNAME=YTD_SALES, ALIAS=ytd_sales, USAGE=I11, ACTUAL=I4,
  MISSING=ON, $
  FIELDNAME=NOTES, ALIAS=notes, USAGE=A200, ACTUAL=A200,
  MISSING=ON, $
  FIELDNAME=PUBDATE, ALIAS=pubdate, USAGE=HYYMDs, ACTUAL=HYYMDs, $

SEGNAME=TITLES, TABLENAME=pubs.dbo.titles, CONNECTION=SQLSRV, KEYS=1, $
```

Relational Efficiencies

Appendix

```
FILENAME=TITLEAUTHOR, SUFFIX=SQLMSS , $
SEGMENT=TITLEAUTHOR, SEGTYPE=S0, $
  FIELDNAME=AU_ID, ALIAS=au_id, USAGE=A11, ACTUAL=A11, $
  FIELDNAME=TITLE_ID, ALIAS=title_id, USAGE=A6, ACTUAL=A6, $
  FIELDNAME=AU_ORD, ALIAS=au_ord, USAGE=I6, ACTUAL=I4,
  MISSING=ON, $
  FIELDNAME=ROYALTYPER, ALIAS=royaltyper, USAGE=I11, ACTUAL=I4,
  MISSING=ON, $

SEGNAME=TITLEAUTHOR, TABLENAME=pubs.dbo.titleauthor, CONNECTION=SQLSRV,
KEYS=2, $
```

Relational Efficiencies

Appendix

```
FILENAME=ROYSCHED, SUFFIX=SQLMSS , $
```

```
SEGMENT=ROYSCHED, SEGTYPE=S0, $
```

```
FIELDNAME=TITLE_ID, ALIAS=title_id, USAGE=A6, ACTUAL=A6, $
```

```
FIELDNAME=LORANGE, ALIAS=lorange, USAGE=I11, ACTUAL=I4,
```

```
MISSING=ON, $
```

```
FIELDNAME=HIRANGE, ALIAS=hirange, USAGE=I11, ACTUAL=I4,
```

```
MISSING=ON, $
```

```
FIELDNAME=ROYALTY, ALIAS=royalty, USAGE=I11, ACTUAL=I4,
```

```
MISSING=ON, $
```

```
SEGNAME=ROYSCHED, TABLENAME=pubs.dbo.roysched, CONNECTION=SQLSRV, KEYS=1,$
```

Relational Efficiencies

Appendix

```
FILENAME=STORES, SUFFIX=SQLMSS , $
SEGMENT=STORES, SEGTYPE=S0, $
  FIELDNAME=STOR_ID, ALIAS=stor_id, USAGE=A4, ACTUAL=A4, $
  FIELDNAME=STOR_NAME, ALIAS=stor_name, USAGE=A40, ACTUAL=A40,
    MISSING=ON, $
  FIELDNAME=STOR_ADDRESS, ALIAS=stor_address, USAGE=A40, ACTUAL=A40,
    MISSING=ON, $
  FIELDNAME=CITY, ALIAS=city, USAGE=A20, ACTUAL=A20,
    MISSING=ON, $
  FIELDNAME=STATE, ALIAS=state, USAGE=A2, ACTUAL=A2,
    MISSING=ON, $
  FIELDNAME=ZIP, ALIAS=zip, USAGE=A5, ACTUAL=A5,
    MISSING=ON, $

SEGNAME=STORES, TABLENAME=pubs.dbo.stores, CONNECTION=SQLSRV, KEYS=1, $
```

Relational Efficiencies

Appendix

FILENAME=SALES, SUFFIX=SQLMSS, \$

SEGMENT=SALES, SEGTYPE=S0, \$

FIELDNAME=STOR_ID, ALIAS=stor_id, USAGE=A4, ACTUAL=A4, \$

FIELDNAME=ORD_NUM, ALIAS=ord_num, USAGE=A20, ACTUAL=A20, \$

FIELDNAME=TITLE_ID, ALIAS=title_id, USAGE=A6, ACTUAL=A6, \$

FIELDNAME=ORD_DATE, ALIAS=ord_date, USAGE=HYMDs, ACTUAL=HYMDs, \$

FIELDNAME=QTY, ALIAS=qty, USAGE=I6, ACTUAL=I2, \$

FIELDNAME=PAYTERMS, ALIAS=payterms, USAGE=A12, ACTUAL=A12, \$

SEGNAME=SALES, TABLENAME=pubs.dbo.sales, CONNECTION=SQLSRV, KEYS=3, \$